What is claimed is:

[Claim 1]   A method of defining recovery and transactional behavior for software.

[Claim 2]   A method of claim 1, further comprising: a method of recovery and transactional behavior for nested, composite Web services using a service composition tool.

[Claim 3]   A method of claim 1, further comprising: a method of declaring the activation and automation of transactional behavior through the association of an attribute to the composite model/program of a set of aggregated and connected software services to indicate activation and automation of transactional behavior for that composite definition at runtime.

[Claim 4]   A method of claim 1, further comprising: a method of declaring transactional behavior through the association of an attribute to a service interface, or a contained service in the context of a containing composite service, to indicate whether the service should be re-invoked upon automatic recovery although it was already invoked successfully in the context of a composite service prior to a system crash or failure.

[Claim 5]   A method of claim 1, further comprising: a method for configurable routing of the flow of execution upon rollback/compensation event through adding a visual port, to the visual presentation of a software service interface as a semantic-based means for routing the flow of execution and activation of another software service, upon a rollback/compensation event within a composite model of a set of connected software services.

[Claim 6]   A method of claim 5, further comprising: declaring whether the built-in rollback behavior of a service embedded within a composite service should be overwritten or extended through the rollback/compensation port.

[Claim 7]   An extensible system through a programmatic interface for the plug-in implementation of non-composite software services with built-in commit/rollback transactional behavior.

[Claim 8]   The system of claim 7, further comprising: a set of generic built-in database access services (e.g. update, query, delete, and add) with automated transactional behavior.

[Claim 9]   An extensible system of automatic transactional execution for nested, composite software services.

[Claim 10]   A system of claim 9, further comprising: a method of built-in transactional behavior for modifications to the data from the inter-process shared memory structures within the transactional execution of composite software services.

[Claim 11]   A system of claim 9, further comprising: a method of providing transactional behavior for any plug-in implementation.

[Claim 12]   A system of claim 9, further comprising: automated implementation for the semantic based transactional behavior defined within a composite service definition.

[Claim 13]   A system of claim 9, further comprising: a method of defining, a context mechanism based on a directed execution graph and an invocation map associated to each composite instance.

[Claim 14]   A method of claim 13, further comprising: a method of step-by-step invocation and automated application of the declared transactional behavior of contained services within a composite service based on it directed execution graph and invocation map.

[Claim 15]   All other methods and systems explicit or implicit in the description of the enclosed invention.